# 实验记录

## 1时间：12：00-21：00

## 2实验目的：

在杰仔的指导下，知道了filtersize设置过小，期待对其调整从而获得更好的实验结果。

## 3实验代码存放文件夹：

2019

## 4实验运行时间：

1天及以上。

## 5实验使用数据信息：

药物作用位点预测，可能存在的药物作用位点标志为1，正样本。其余其他位点均为负样本。消除位于边界的负样本，包含正样本位点的负样本。

## 6.1，数据集正负样本设置，比例设置，

正负样本1：33。采用十则交叉验证模式构建10个数据集的数据，依次进行学习。目前没完成整体评估+存取模型+选取最优模型等步骤

## 6.2，网络设置：

if compiletimes == 0:

input = Input(shape=(input\_row, input\_col))

filter1 = 16

filtersize1 = 2

dropout1 = 0.25

L1CNN = 0

nb\_classes = 2

batch\_size = batch\_size

actfun = "relu";

nadam = Nadam(lr=0.00001)

optimization = nadam

dense\_size1 = 128

dense\_size2 = 64

dense\_size3 = 8

dropout\_dense1 = 0.298224

dropout\_dense2 = 0

dropout\_dense3 = 0

input = Input(shape=(input\_row, input\_col))

x = conv.Conv1D(filter1, filtersize1, init='glorot\_normal', W\_regularizer=regularizers.l2(L1CNN),

border\_mode="same")(input)

x = Dropout(dropout1)(x)

x = Activation(actfun)(x)

x = core.Flatten()(x)

output = x

output = Dropout(dropout1)(output)

output = Dense(dense\_size1, init='glorot\_normal', activation='relu')(output)

output = Dropout(dropout\_dense1)(output)

#output = Dense(dense\_size2, activation="relu", init='glorot\_normal')(output)

#output = Dropout(dropout\_dense2)(output)

output = Dense(dense\_size3, activation="relu", init='glorot\_normal')(output)

output = Dropout(dropout\_dense3)(output)

out = Dense(nb\_classes, init='glorot\_normal', activation='softmax')(output)

cnn = Model(input, out)

cnn.compile(loss='binary\_crossentropy', optimizer=optimization, metrics=['accuracy'])

之后使用输出模式查看网络。

## 7实验变量：

网络设置中的filter1：分别设置为16，32，48，64。

## 8实验运行记录

## ，，，，。

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | 开始时间 | 输出文件 | 结束时间 | GPU编号（数量） | 运行id | kernalsize |
| 1 | 下午 |  |  | 4 | 32452 | 64 |
| 2 | 下午 |  |  | 5 | 32013 | 48 |
| 3 | 下午 |  |  | 6 | 26526 | 32 |
| 4 | 下午 |  |  | 7 | 17859 | 16 |

## 

## 9实验结果存放位置

将图片下载后存储

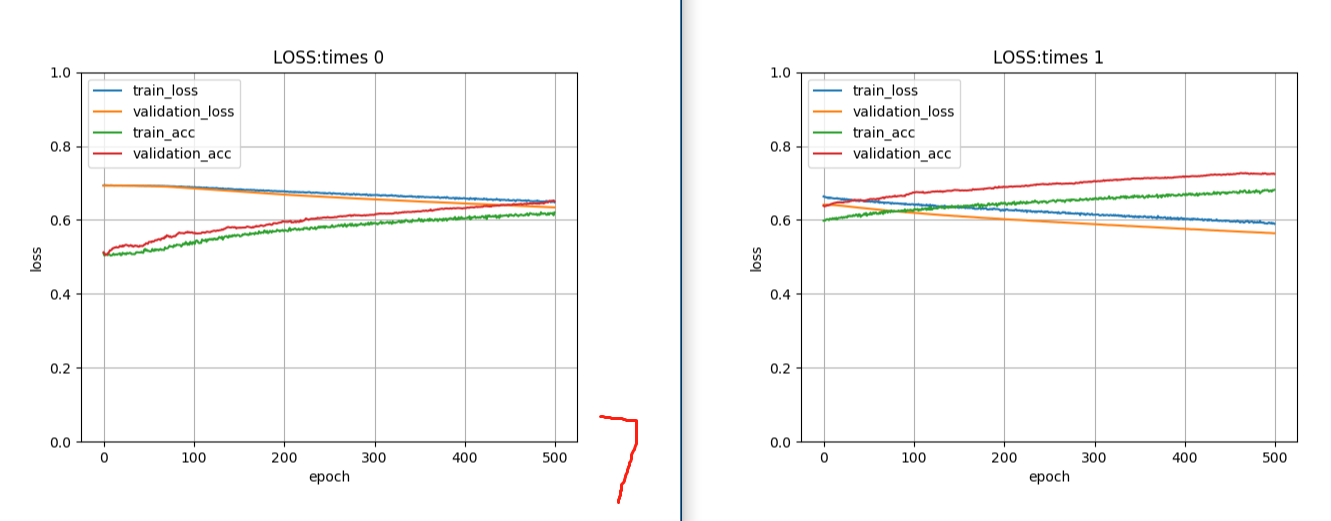
## 10结果分析

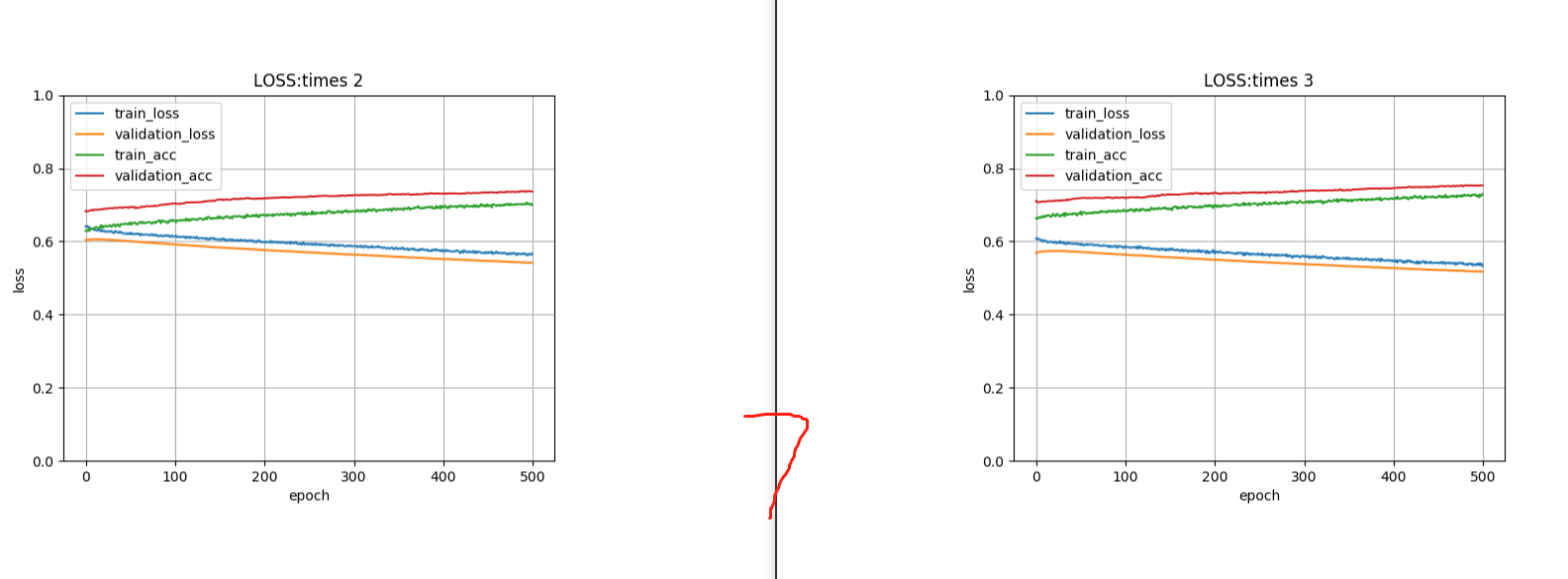
分别获取实验的第0-3，30-33图进行比较。

结果分析1，acc+loss

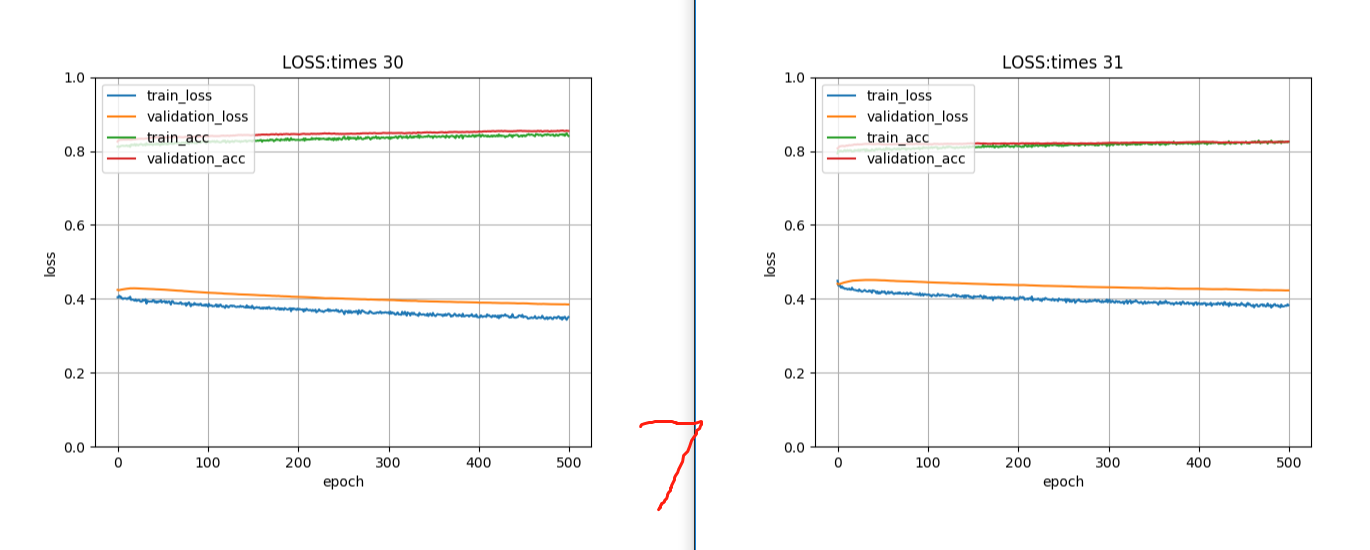
【7号，size = 16】

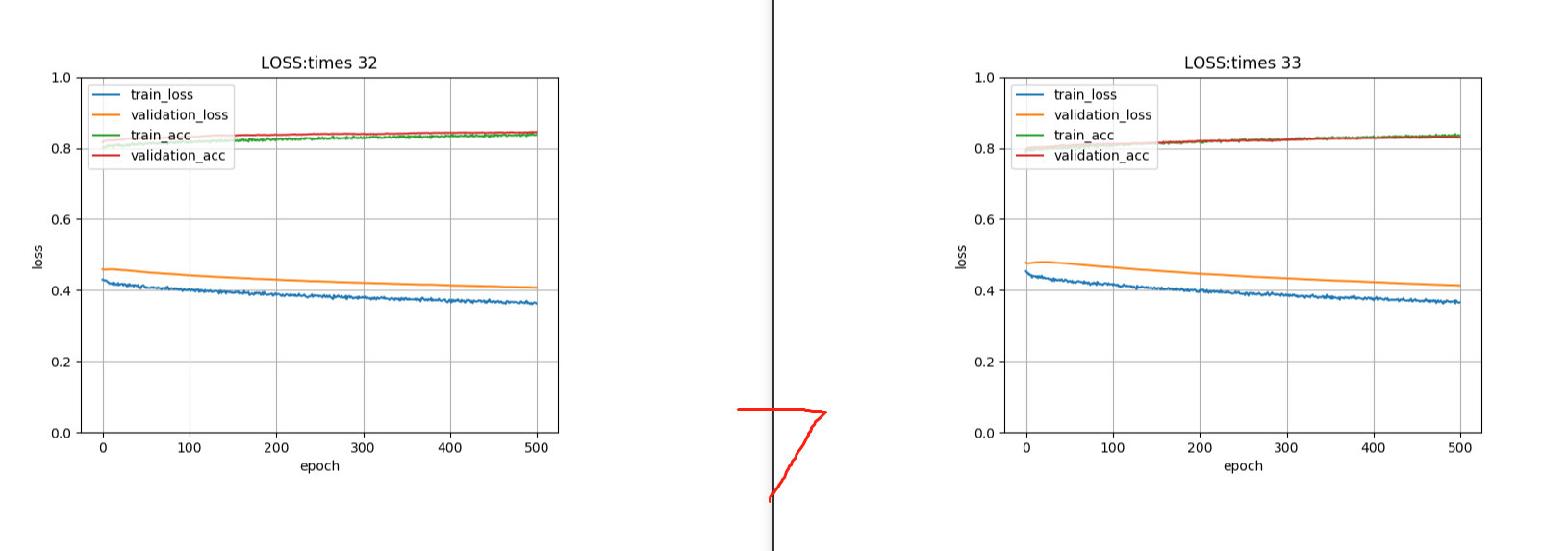
0-3





30-33



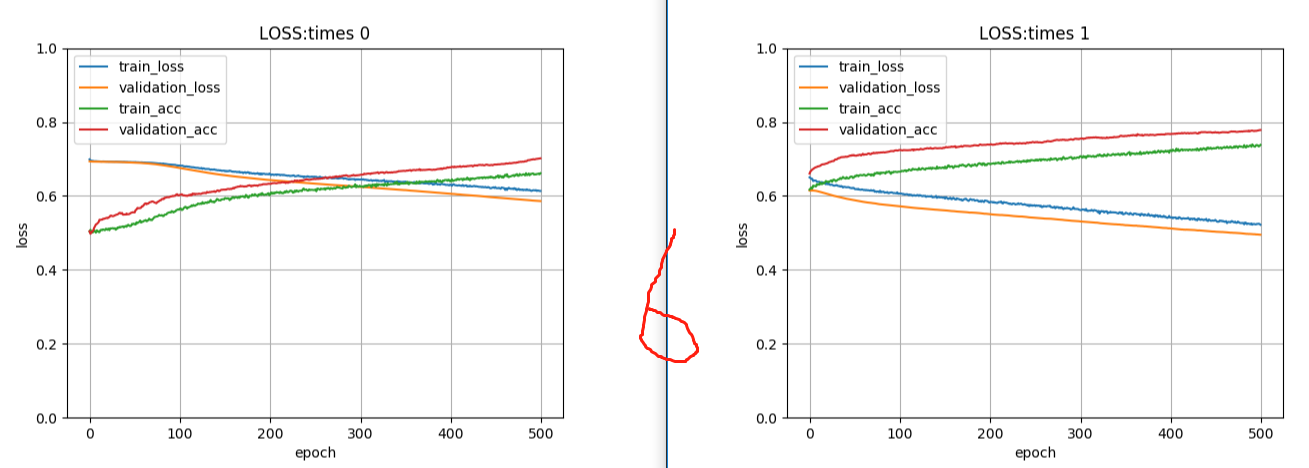


特点总结：小的fitersize虽然没有获得更高的acc和更低的loss,但是可以获得更好的训练集和验证集拟合度。

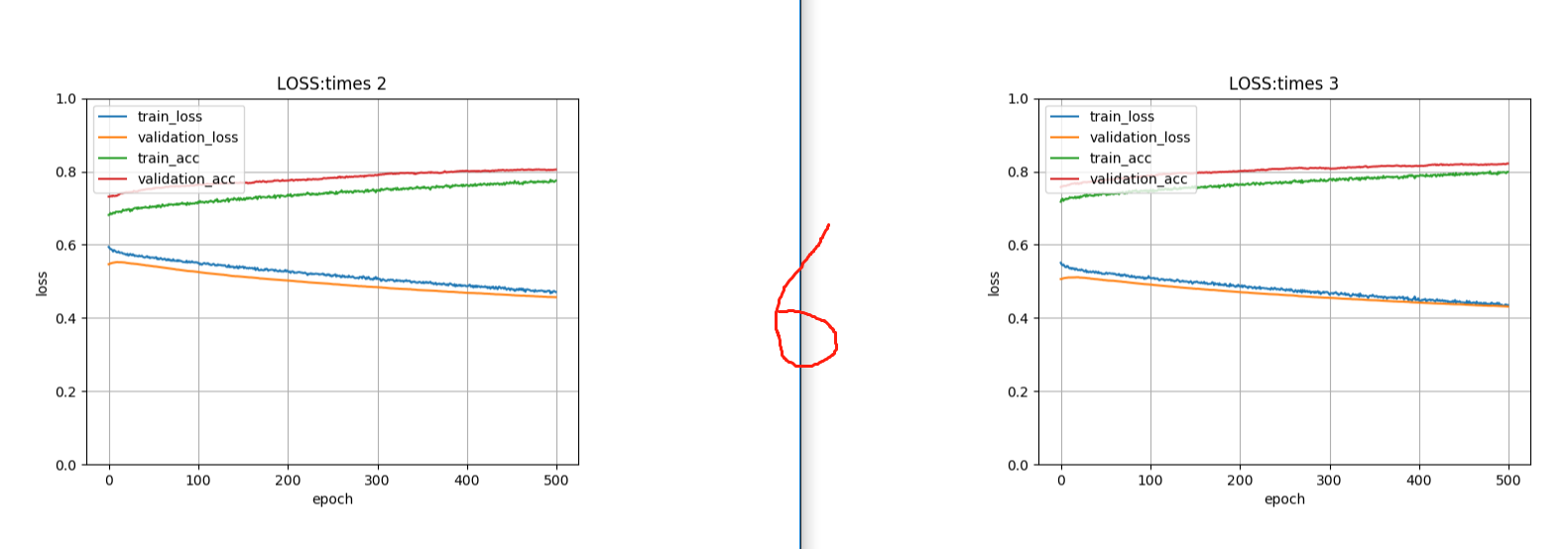
===================================================================

【6号，size = 32】

(0-3)



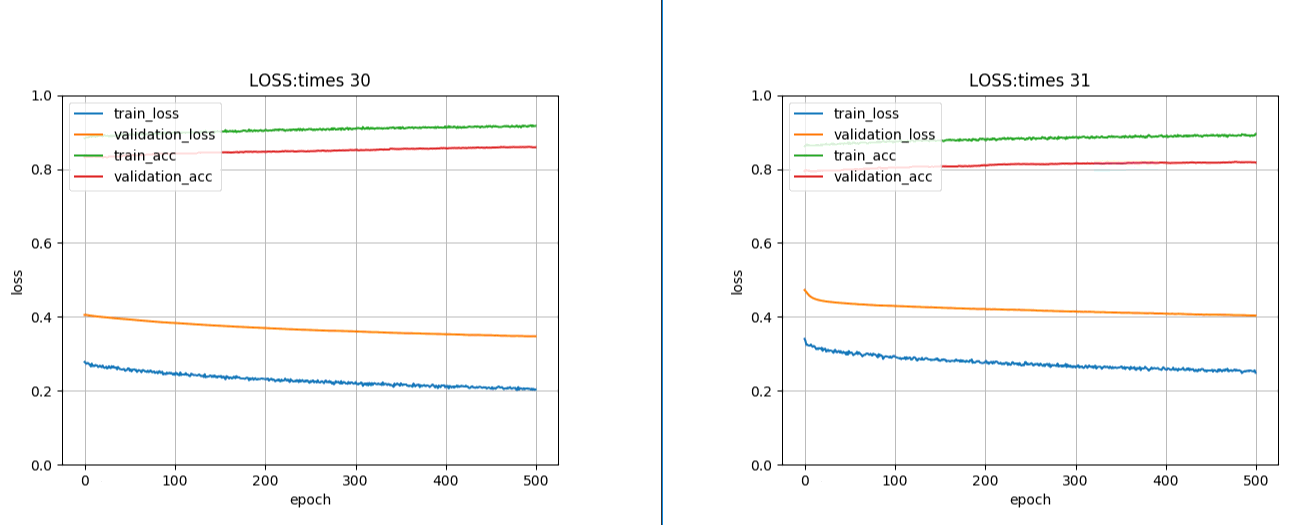
（6-7对比）从0号图中，我们发现，size大的在开始阶段acc提升更高一点。(size32的acc高，loss低)

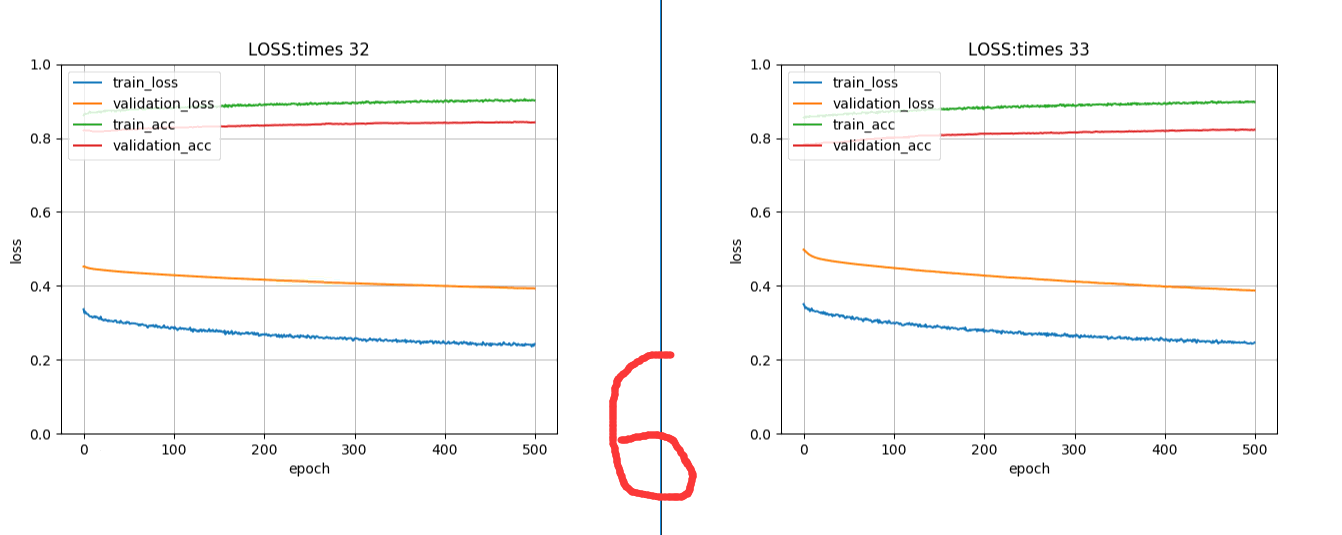


（6-7对比）从3号图中，我们可以发现，size大的loss曲线贴合更好，loss下降更快，acc曲线训练集与测试集相差较大，验证集偏高。（acc测试集与验证集在第七组数据的时候重合，之后训练集偏高5%-10%）

（loss测试集与验证集在第四-五组数据的时候重合，之后训练集偏低10%-15%）

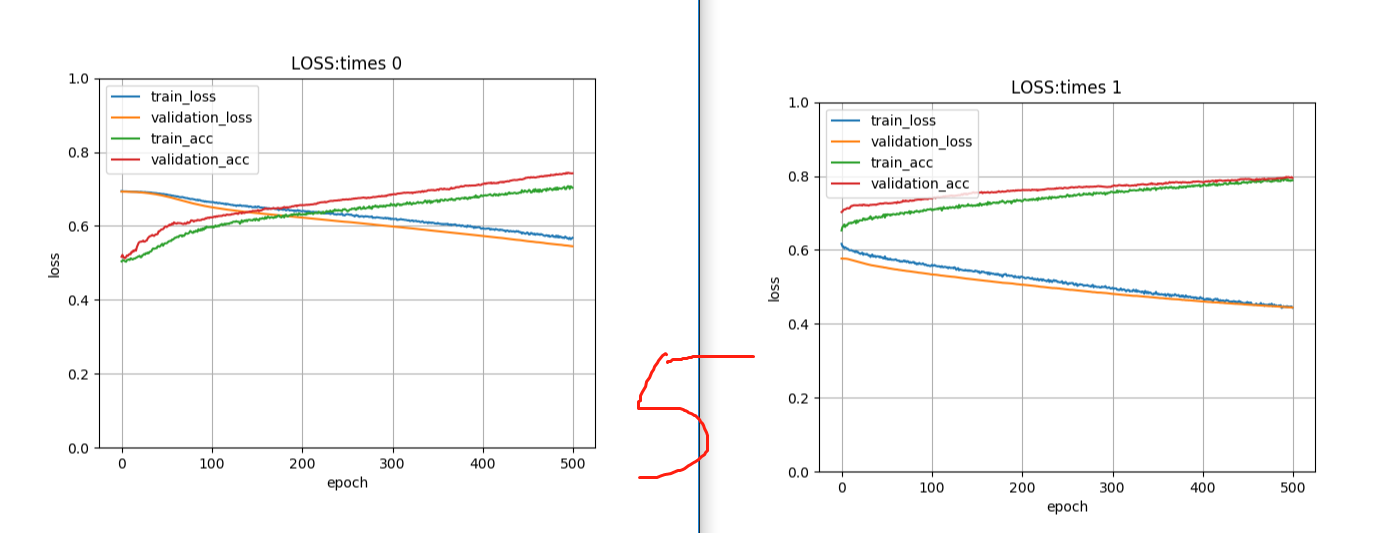
（30-33）最后这部分图线说明，train和val，有较大差别。其中loss的差值最大，目前在15%左右。但远比7号对照组差值大。说明：filtersize增大，将会增大loss的训练集与验证集数值的差值。而train\_loss较低，说明？目前不知道。



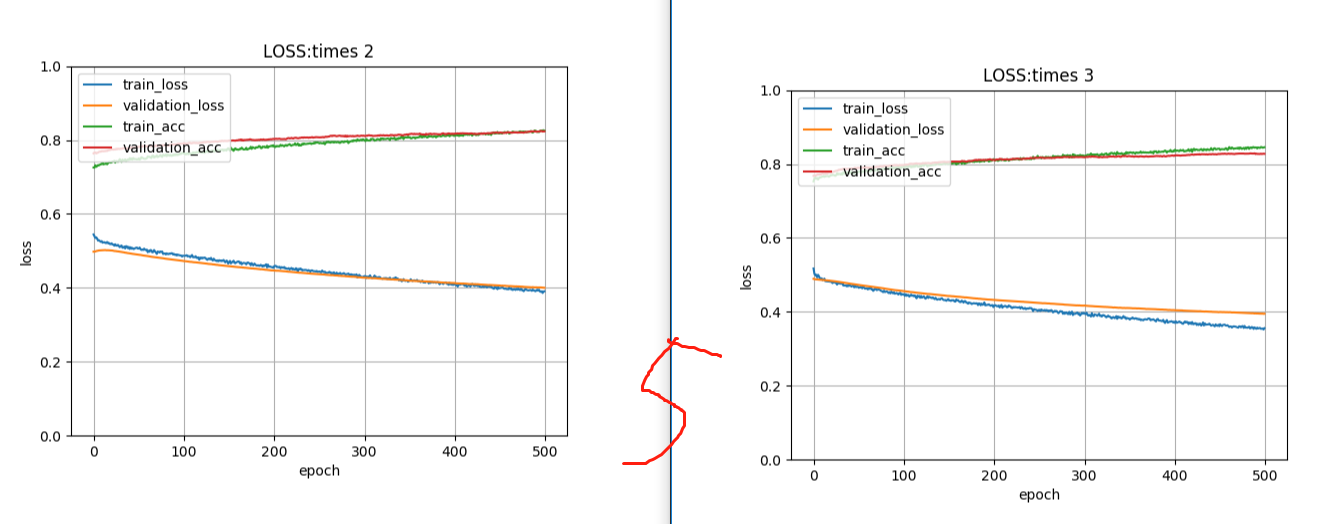


【5号，size = 48】

（0-3）

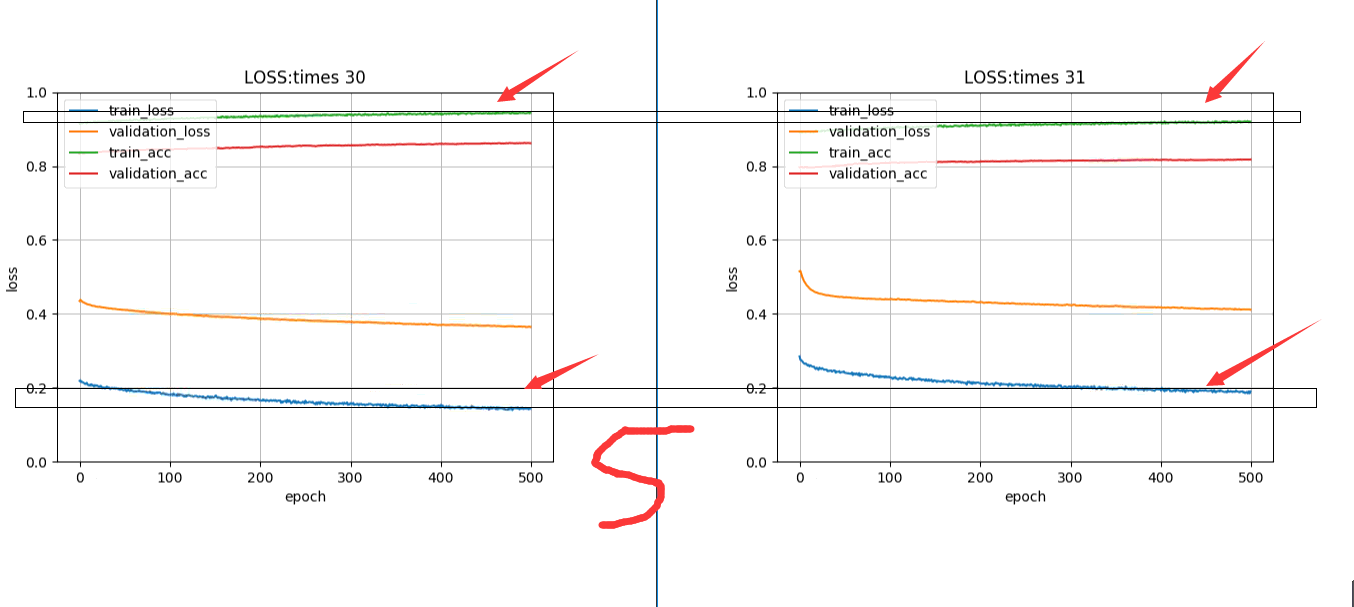


（对比7-6-5的0号图）随着size的增大，我们发现训练得到更好的结果的过程提前了。交叉时间从第500次epoch变成了400次，200次。虽然交叉没有意义，但是可以说明曲线斜率增大了。同时发现，acc与loss的训练集与验证集也提前拟合，并在之后分开。

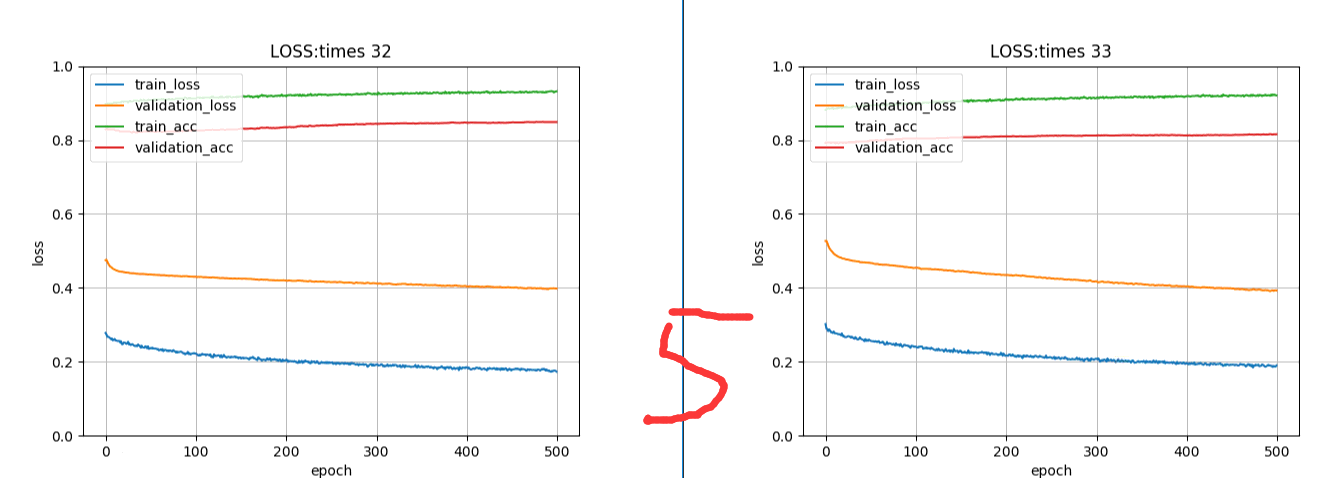


如上图所示，在5号中，他们在第3-4组数据就完成了高度拟合并分开的过程。

（30-33）

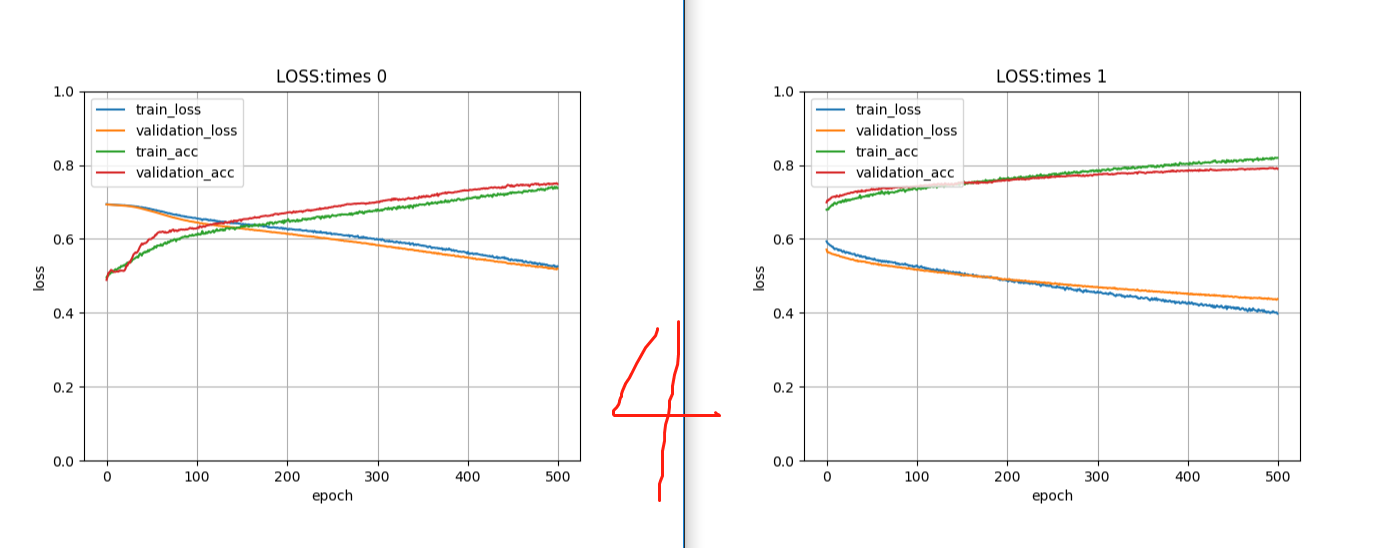


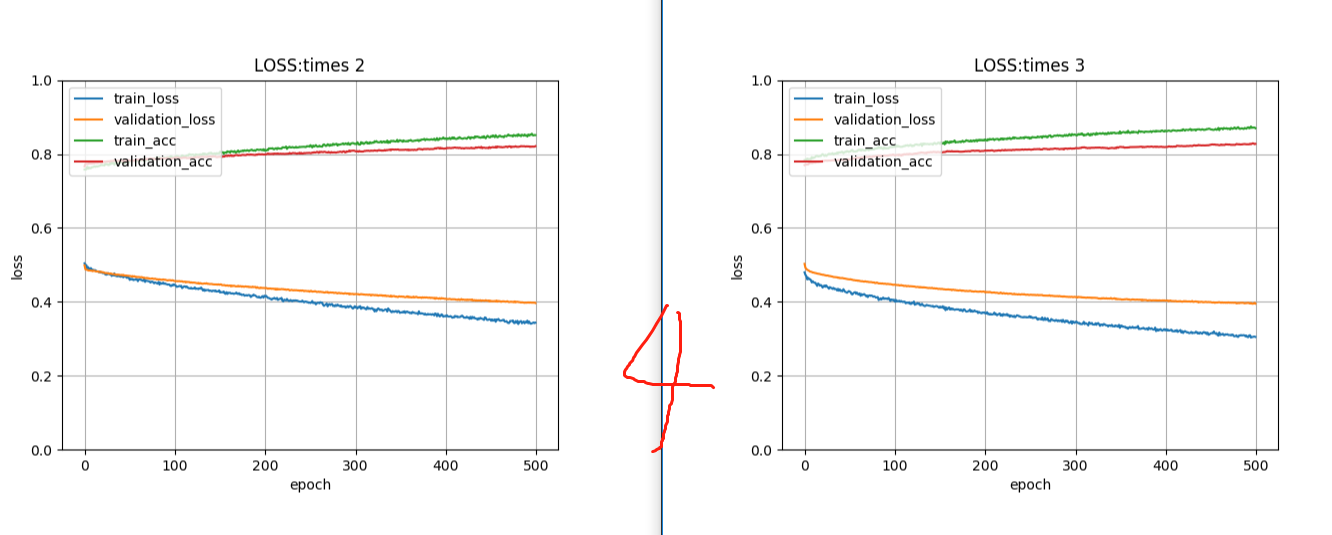
在30-31图中，观察到后一组曲线相对前一组曲线逆趋势变化的现象。可能在4号训练结果中出现类似情形。



此图证明他们来自一数据训练结果，而不是多组。

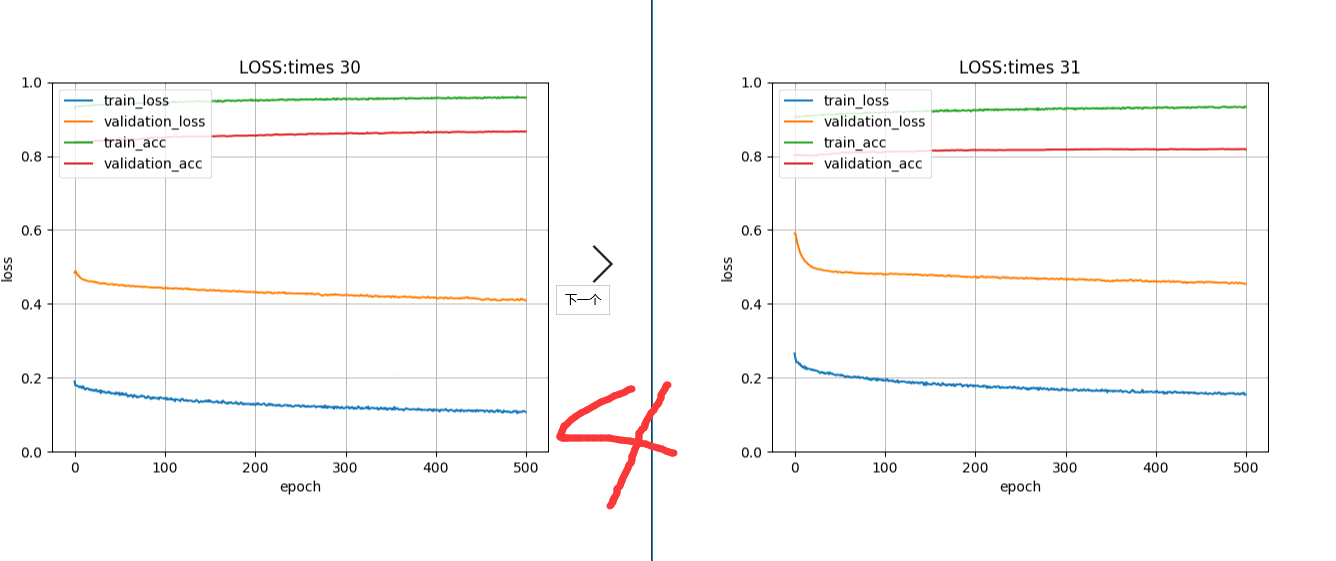
【4号，size = 64】

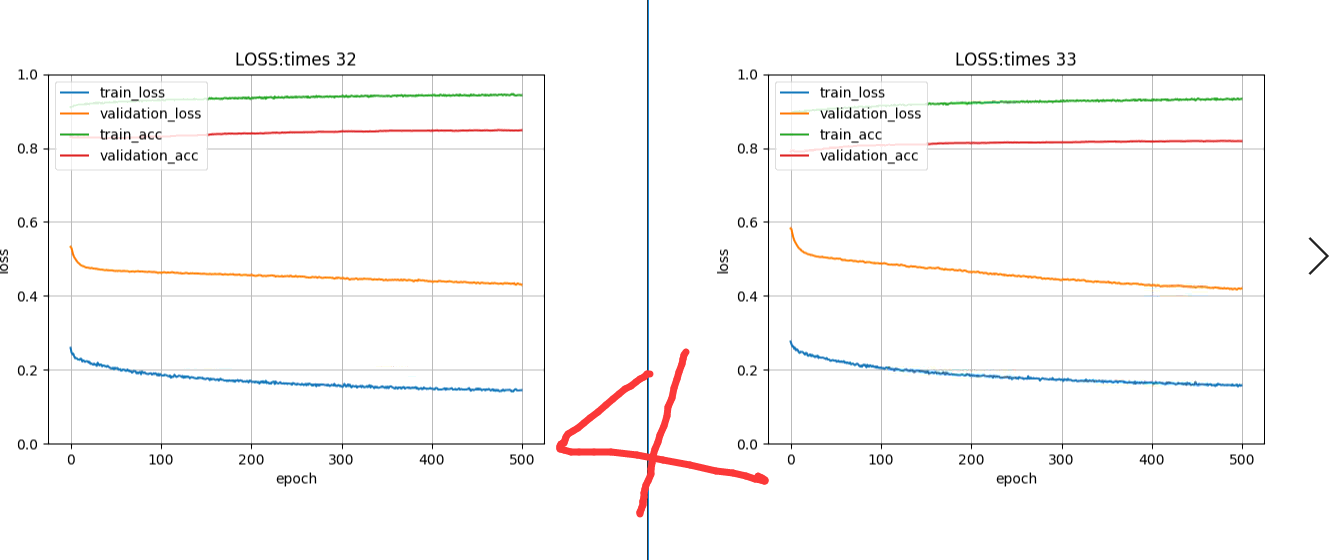




也能说明上述观点

（30-33）

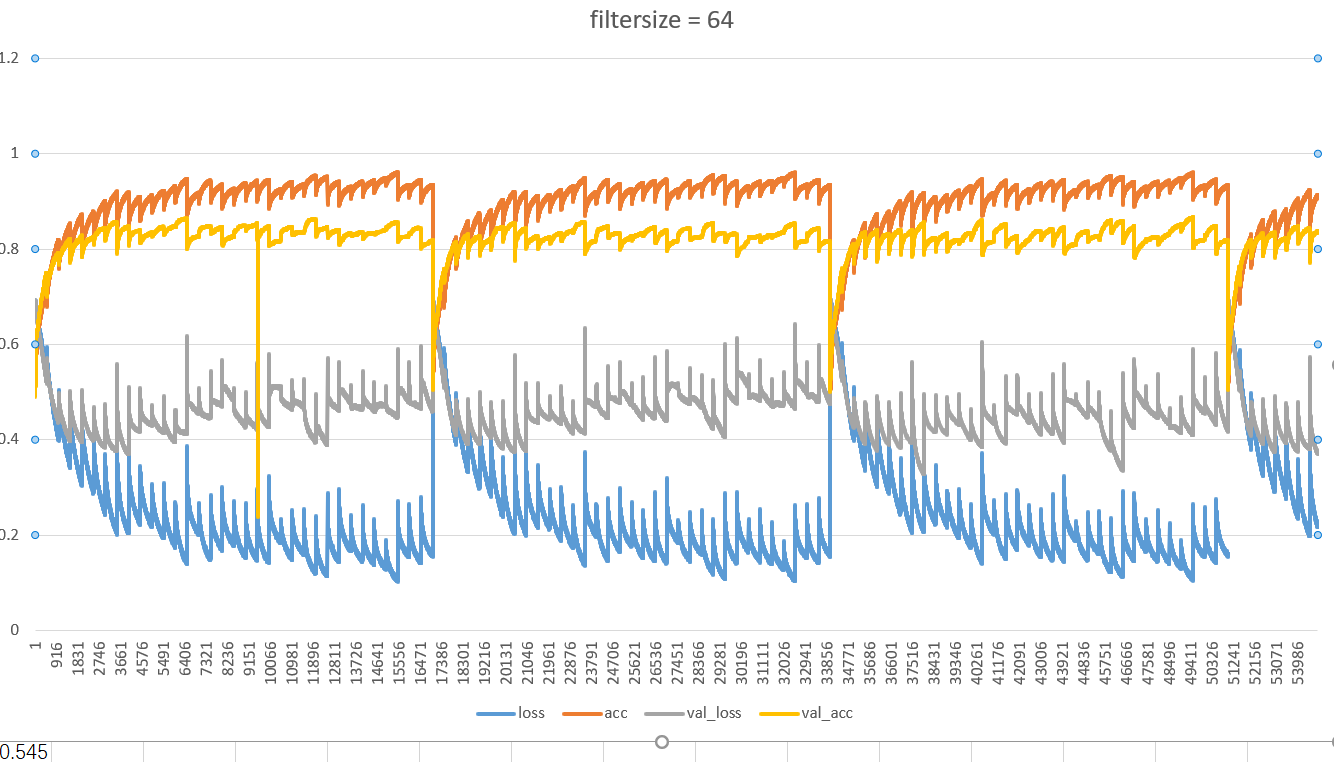
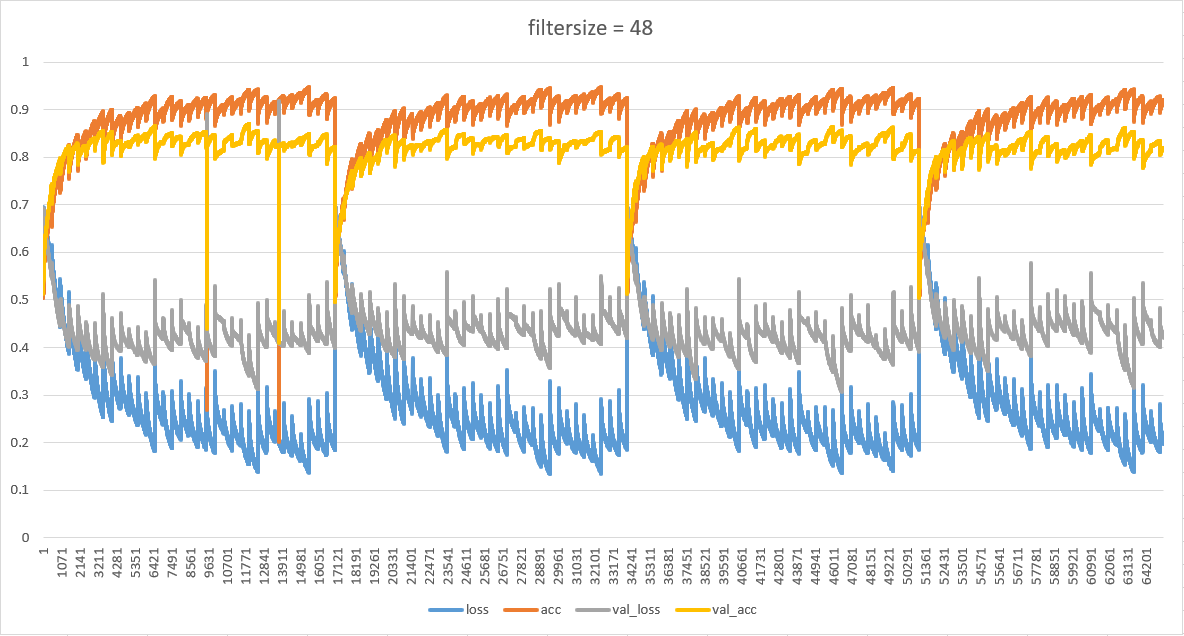




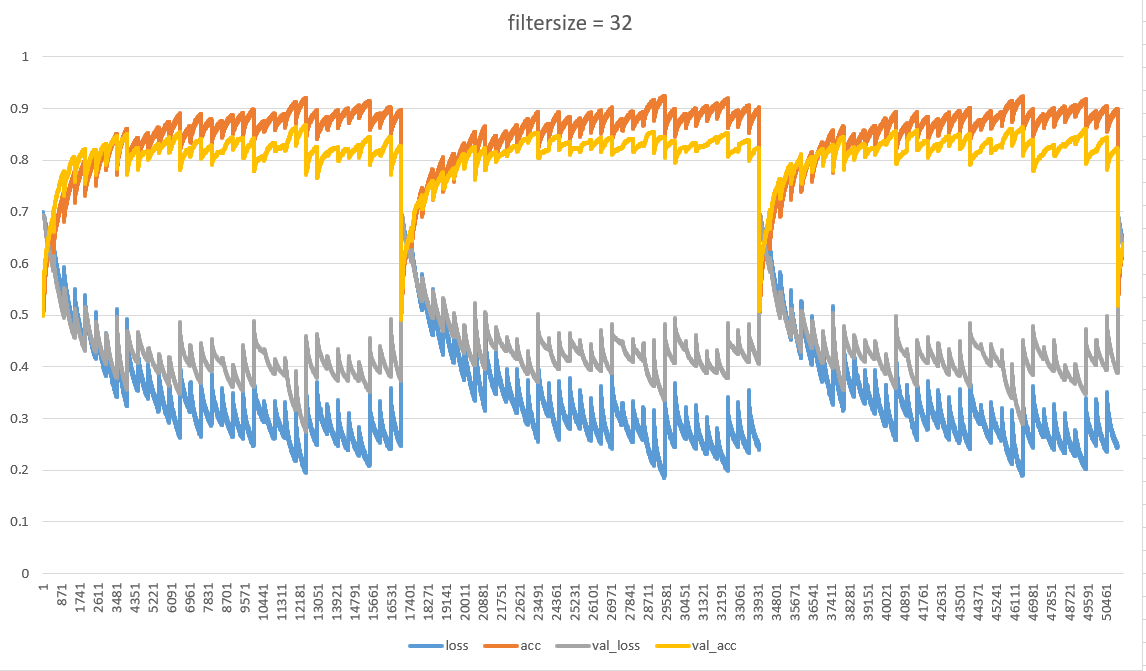
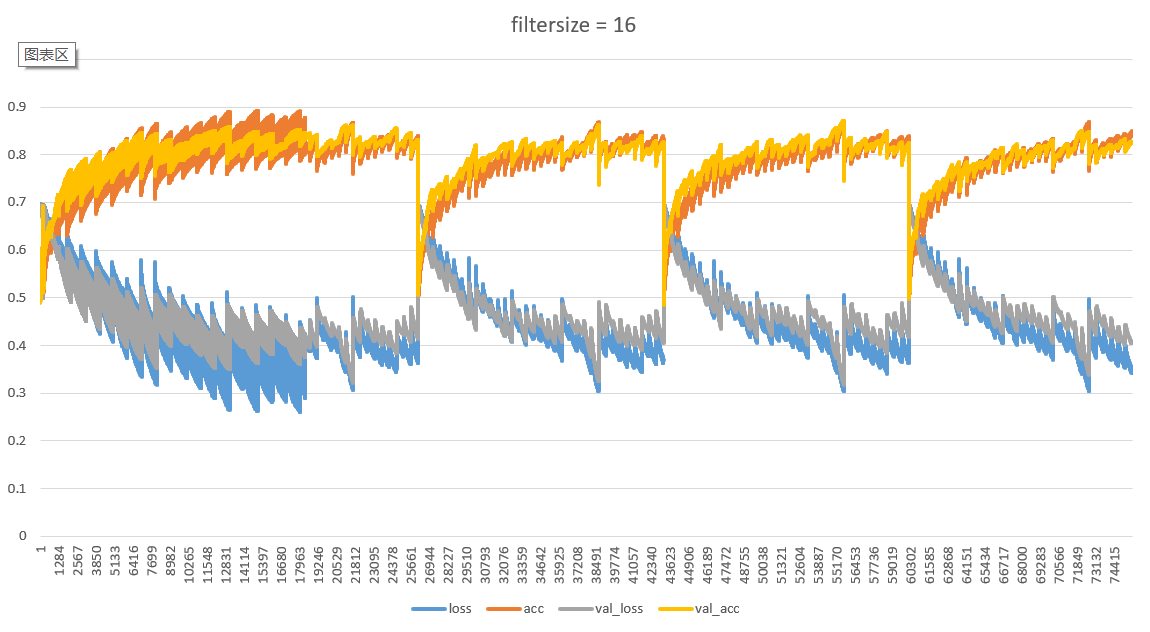
Loss的插着增大，从小于20%-🡪20%🡪大于20%

获取到每个实验的所有loss,val\_loss,acc,val\_acc,并且作图。十折交叉运算中只运行了前三-四组数据，所以在图线上只显示了3-4组acc和loss曲线显示出来显示的漏斗。

【特别注释】把acc和loss画在一张图上目的是看一张图，看到两种曲线，没有别的特殊用途，理论上也不应该这么做。那么，让我们观察一下4组不同的曲线吧！

四幅图分别是filtersize为64，48，32，16这四个数值的结果。

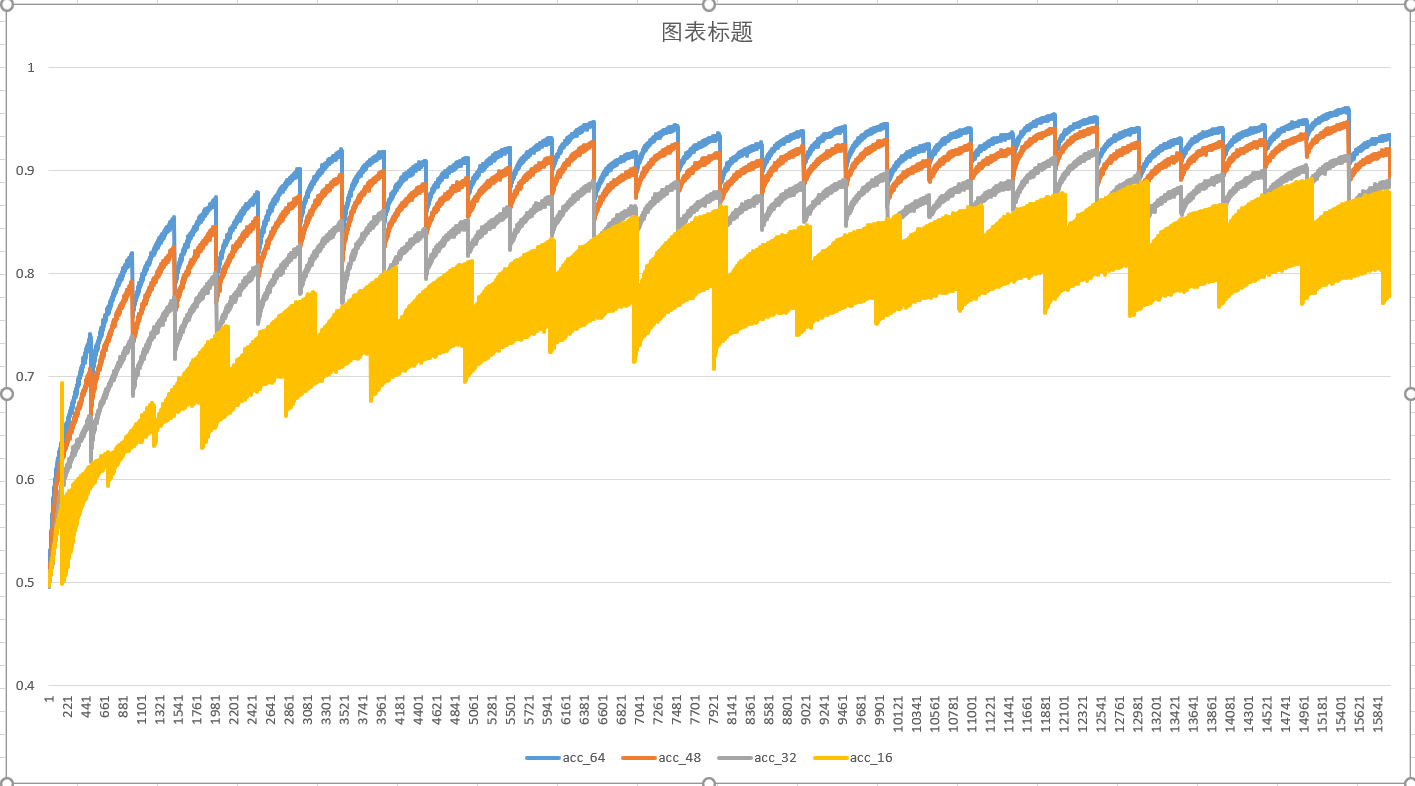
 

结论1：随着filtersize增大，acc,val\_acc的数值均逐渐增大。

结论2：filtersize在48，64这两组数据中，均出现了2~3次异常（可能因为数据导入的问题）

结论3：随着filtersize增大，loss,val\_loss也降低得更低了。

我们单独比较acc的数值，如下图所示（消除掉特殊数据点），扩大了纵坐标。



首先我们可以看到每个曲线的小尖刺部分说明，每一轮迭代都是增加了acc,也就是提升了成绩。

其次我们大概可以看到有33个分段，代表着每次正样本与1-33个负样本的配对。

最后我们发现，随着filtersize的增大，acc有所提高，也就是准确率提升了！

从图中可以看出，前三分之一部分曲线是在增加的，后边几乎就是波动，没有什么增加的趋势，仅仅简单的起起伏伏。由此宫师姐表示，要把nbclass设置为15，从而节省一半训练时间。另一个发现是：随着filtersize的增加，曲线训练处最好acc的时间越早。

最下面一条曲线filtersize = 16,有重叠部分，所以看起来不像是曲线。

## 11总结